

APPLICATION
FOR
UNITED STATES LETTERS PATENT

TITLE: METHOD AND APPARATUS FOR CALL
PROCESSING

APPLICANT: ANDREW J. PROKOP

Express Mail No.: EL542038195US

Date: JUNE 7, 2000

Prepared by: Trop, Pruner & Hu, P.C.
8554 Katy Freeway, Ste. 100, Houston, TX 77024
713/468-8880 [Office], 713/468-8883 [Fax]

METHOD AND APPARATUS FOR CALL PROCESSING

BACKGROUND

The invention relates to a method and apparatus for call processing.

Packet-based data networks are widely used to link various types of network
5 elements, such as personal computers, servers, gateways, network telephones, and so
forth. Data networks may include private networks (such as local area networks or
wide area networks) and public networks (such as the Internet). Popular forms of
communications between network elements across packet-based data networks
include electronic mail, file transfer, web browsing, and other exchanges of digital
10 data.

With the increased capacity and reliability of packet-based data networks,
voice communications (including telephone calls, video conferencing, and so forth)
over data networks have become possible. Voice communications over data networks
are unlike voice communications in a conventional public switched telephone network
15 (PSTN), which provides users with dedicated, end-to-end circuit connections for the
duration of each call. Communications over data networks, such as IP (Internet
Protocol) networks, are performed using packets or datagrams that are sent in bursts
from a source to one or more destination nodes. Voice data sent over a data network
typically shares network bandwidth with conventional non-voice data (e.g., data
20 associated with electronic mail, file transfer, web access, and other traffic).

Various standards have been proposed for voice and multimedia
communications over data networks. One such standard is the H.323
Recommendation from the International Telecommunications Union (ITU), which
describes terminals, equipment, and services for multimedia communications over
25 data networks. Another standard for voice and multimedia communications is the
Session Initiation Protocol (SIP), which establishes, maintains, and terminates
multimedia sessions over a data network. SIP is part of multimedia data and control
architecture developed by the Internet Engineering Task Force (IETF). The IETF
multimedia data and control architecture also includes other protocols to enable voice
30 and multimedia sessions over data networks.

To establish a call, an originator device sends a call request to a destination
device, with the call request containing an identifier of the destination device. The

identifier may be a telephone number, a SIP address, or some other identifier. In a network having multiple network elements capable of participating in a voice or multimedia session, each network element is typically assigned a unique identifier. In a SIP network, one or more SIP proxies may be used to resolve locations of devices associated with a logical identifier. Thus, for example, a call request containing a telephone number may be processed by the SIP proxy, with the telephone number resolved to the physical location of the destination device.

Although improved technology has enhanced the flexibility in how a call session is established, some limitations still exist. For example, in a conventional network, convenient mechanisms have not been provided to enable a group of network elements to be associated with a single or a reduced number of logical identifiers for call sessions over packet-based networks. A need thus exists for a method and apparatus to enable more flexible techniques and mechanisms for call processing over packet-based data networks.

SUMMARY

In general, according to one embodiment, a method of providing call processing in a communications system having a packet-based network comprises receiving a first call request from a first device and sending a response to the first call request to indicate an attempt to establish a call session. One of plural destination devices to contact is identified in response to the first call request, and a second call request is sent to the one destination device.

In general, according to another embodiment, a system capable of participating in call sessions over a packet-based network includes a first module adapted to process a first call request from a first device in a server mode and a second module adapted to send a second call request to a second device in a client mode in response to the first call request. Further, a third module is adapted to process at least one message from one of the first and second devices in a proxy mode.

Some embodiments of the invention may have one or more of the following advantages. Increased capabilities are provided for management of inbound calls targeted to a community of network entities. Decisions may be made on which of plural entities are accessed in response to a given inbound call or in response to input data received during a call. In one example application, a single or a small number of

telephone numbers may be presented to the outside world while, within the community, multiple parties (greater than the number of telephone numbers presented to the outside world) are present that can take the inbound call. By combining tasks performed by various logical entities (e.g., client, server, proxy) for call processing in a packet-based network, flexibility may be enhanced.

Other features and advantages will become apparent from the following description, from the drawings, and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of an embodiment of a communications system including a packet-based network.

Fig. 2 is a block diagram of components of a call processing control (CPC) system in accordance with an embodiment usable in the communications system of Fig. 1, the CPC system providing combined server, client, and proxy modules.

Fig. 3 is a message flow diagram of various states of a call session involving a user system and the CPC system in accordance with an embodiment.

Fig. 4 is a state diagram of processing performed in the CPC system of Fig. 2.

DETAILED DESCRIPTION

In the following description, numerous details are set forth to provide an understanding of the present invention. However, it will be understood by those skilled in the art that the present invention may be practiced without these details and that numerous variations or modifications from the described embodiments may be possible. For example, although reference is made to a Session Initiation Protocol (SIP) in the described embodiments, other embodiments may include other protocols for real-time interactive call sessions.

Referring to Fig. 1, a communications system 10 includes a packet-based data network 12 and various network elements. The network elements may include a user system 16 and proxies 14 and 18, as well as various network elements in a community 26. As used here, a “community” may refer to any collection or group of systems or devices that may be grouped together. For example, the community 26 may include an enterprise, such as a corporation or other organization.

The community 26 contains a call processing control (CPC) system 20 in accordance with an embodiment that includes combined logical entities. Such logical

entities may include a client task, a server task, and a proxy task. Under control of the client, server, and proxy tasks, the CPC system 20 is capable of processing inbound calls over the data network 12 and routing the calls to appropriate ones of various network elements (in the community 26) that may be coupled to a local network 32, which may be a packet-based network. As examples, the network elements include an IVR (integrated voice response) system 22 to provide integrated voice services, an announcement system 24, and various systems 28, 30 associated with plural agents. Agents may be persons associated with a customer service department, technical support personnel, and so forth, as examples. Other network elements may also be connected to the local network 32.

One capability provided by the CPC system 20 is the ability to present one or a relatively small number of telephone numbers or other identifiers to the outside world (outside the community 26). A call originated from an outside network element and directed to a telephone number or address associated with the community 26 is processed by the CPC system 20. The CPC system 20 accepts the call and determines, based on information in the call request or further input or information from the calling entity, which of the network elements in the community 26 is to be involved in the call session. In effect, the CPC system 20 is able to perform one-to-many address or identifier translation of an address or identifier contained in an inbound call request. In addition, in an established same call session, the CPC system 20 can direct a call from a first device in the community 26 to another device in the community.

Any one of the network elements in the community 26 may place an outgoing call through the CPC system 20. A called party on the outside world may receive an indication that the originator is the community 26 rather than one of the individual agents 28, 30. The CPC system 20 thus provides a combination of tasks: a server to accept a call from a client; a proxy to process a call on behalf of another network element; and a client to initiate a call to a destination device inside or outside the community 26.

As used here, a "call session" refers generally to either an audio (e.g., voice), a video, or a multimedia session established between two or more network elements (and parties using those elements) coupled to the data network 12 (or any other data network). A call session may also be a text-based chat session, such as an instant messaging session. As used here, an "interactive" call session refers to a call session

in which two or more parties are involved in an exchange of voice and/or video data (or text data) in an established session between two or more network elements. A “real-time” interactive call session refers to an exchange of data, such as audio and/or video data (or text data), on a substantially real-time basis between two terminals. A session is substantially real time if interaction is occurring between two end points or parties, with a communication from one end point followed relatively quickly by a response or another communication from the other end point, typically within seconds, for example. A “call request” is a message (inbound or outbound) generated to establish a call session.

Interactive call sessions are contrasted with electronic mail messaging, for example, in which a first participant sends a message over a data network to a second participant. No indication is usually provided back to the first participant that the second participant has received the message or that the second participant is even at his or her terminal. In contrast, an interactive session involves a request followed by some acknowledgment that a called party has accepted the call request. This enables the interactive session in which participants exchange data (e.g., voice, video, and/or text).

As used here, a “data network” or “network” may refer to one or more communications networks, channels, links or paths, and systems or devices (such as routers) used to route data over such networks, channels, links, or paths. Packet-based data networks communicate with packets, datagrams, or other units of data over the data networks. Unlike circuit-switched networks, which provide a dedicated end-to-end connection or physical path for the duration of a call session, a packet-based network is one in which the same path may be shared by several network elements.

A packet-based network may be a packet-switched network, such as an Internet Protocol (IP) network, which is based on a connectionless internetwork layer. Packets or other units of data transmitted into a packet-switched data network may travel independently over any path (and possibly over different paths) to a destination point. The packets may even arrive out of order. Routing of the packets is based on one or more addresses carried in each packet.

One version of IP is described in the Request for Comments RFC 791, entitled “Internet Protocol,” dated September 1981. Other versions of IP, such as IPv6, or other connectionless packet-switched standards may also be utilized in further

embodiments. A version of IPv6 is described in RFC 2460, entitled "Internet Protocol, Version 6 (IPv6) Specification," dated December 1998.

Packet-based networks may also be connection-oriented networks, such as Asynchronous Transfer Mode (ATM) or Frame Relay networks. In a connection-oriented packet-based network, a virtual circuit or connection is established between two end points. In such connection-oriented networks, packets are received in the same order in which they were transmitted.

One protocol that provides for establishment of streaming communications includes a Session Initiation Protocol (SIP). SIP is part of the multimedia data and control architecture from the Internet Engineering Task Force (IETF). A version of SIP is described in RFC 2543, entitled "SIP: Session Initiation Protocol," dated 1999. SIP may be used to initiate call sessions as well to invite members to a session that may have been advertised by some other mechanism, such as electronic mail, newsgroups, web pages, and other mechanisms. The other protocols in the IETF multimedia and control architecture include the Resource Reservation Protocol (RSVP), as described in RFC 2205, for reserving network resources; the Real-Time Transport Protocol (RTP), as described in RFC 1889, for transporting real-time data and providing quality of service (QoS) feedback; the Real-Time Streaming Protocol (RTSP), as described in RFC 2326, for controlling delivery of streaming media; the Session Description Protocol (SDP), as described in RFC 2327; and the Session Announcement Protocol (SAP), for advertising multimedia sessions.

Other standards may be employed in further embodiments for controlling call sessions over the data network 12. Such other standards may be any other standard that provides for interactive, real-time streaming communications over the data network 12. One alternate standard is the H.323 Recommendation from the International Telecommunication Union (ITU).

In accordance with SIP, the various elements may be SIP clients, SIP servers, or SIP proxies. In the case of one embodiment of the CPC system 20, an entity having a combination of a SIP client, SIP server and SIP proxy may be provided. A SIP client system includes a client application program that is capable of sending SIP requests to perform call requests. A SIP server system includes a server application program that accepts SIP requests to serve as calls and to send back responses to SIP requests. Any user or subscriber system may be a SIP client system when making calls and a SIP server system when receiving calls. A SIP proxy system includes an

intermediary program that acts as both a server and a client for making requests on behalf of other clients.

Thus, in the example of Fig. 1, the user system 16 and any one of the agent systems 28, 30 may be SIP client or SIP server systems. When establishing a call, a SIP client system may establish a call directly over the data network 12 with a server or destination system. Alternatively, the SIP client system may establish the call through one or more SIP proxies.

A call session over the data network 12 may also involve a terminal coupled to a traditional circuit-switched network, such as a public switched telephone network (PSTN) 42. The PSTN-connected device may be a standard telephone 44. Circuit-switched signaling and traffic are translated to packet-based signaling and traffic by a PSTN gateway 40. Thus, a user at the telephone 44 may dial the one or more telephone numbers associated with the community 26 to reach the CPC system 20. Similarly, any one of the network elements in the communications system 10 may direct a call targeted to the telephone 44.

Referring to Fig. 2, the components of the CPC system 20 in accordance with one embodiment are illustrated. The CPC system 20 includes a call processing module 100 that may include several elements, including a proxy module 102, a user agent client (UAC) module 104, and a user agent server (UAS) module 106. Although indicated as sub-modules within a general module, the proxy, UAC and UAS may actually be separate modules in the CPC system 20 in a further embodiment.

As used here, the client, server, and proxy modules may also be referred to as tasks. In addition, the CPC system 20 can be said to be operating in one of client, server, or proxy modes depending on which of the client, server, or proxy modules or tasks have control.

The call processing module 100 may also include a call state script 108 that determines what the call processing module 100 should do next in the establishment, management, or termination of a call session. In one embodiment, the call state script 108 may be implemented as an Extensible Markup Language (XML™) script. One version of XML™ is described in "Extensible Markup Language (XML™) 1.0," Worldwide Web Consortium (W3C) Recommendation, dated February 1998. XML™ allows one to define a customized markup language for many classes of documents.

XML™ may be used to encode may different types of information, including the script modules as described herein for creating telephony services in accordance with some embodiments.

5 The CPC system 20 includes a network interface, which may be in the form of a network interface card or controller, coupled to the data network 12 and local network 32. Above the network interface 114 are device drivers and transport and network layers 112. The transport and network layers 112 may include a TCP/IP stack and/or a UDP/IP stack. TCP is described in RFC 793, entitled "Transmission Control Protocol," dated September 1981; and UDP is described in RFC 768, entitled,
10 "User Datagram Protocol," dated August 1980. TCP and UDP are transport layers for managing connections between network elements over an IP network. For processing of SIP control messages, a SIP stack 110 is provided. The SIP stack 110 is basically a state machine to parse SIP messages.

15 The various software routines or modules in the CPC system 20 may be executable on one or more control units 118. Instructions of such software routines or modules may be stored in one or more storage devices 120 and loaded for execution on the one or more control units 118.

20 One example use of the CPC system 20 is in a call center, which may be the technical support department or customer service department of a company that has many agents answering calls. In the call center example, the community 26 is the call center. The call center may present a single or small number of telephone numbers or other call identifiers to the outside world. A customer calls one of these numbers or identifiers and is given some ringback indication or an announcement. The call center software, which may be the call processing module 100 of the CPC system 20, may
25 connect the caller to a variety of destinations. For instance, a caller might be first connected to the IVR system 22 (Fig. 1) to collect input data from the customer. The collected data is then used to determine which agents may be able to meet the customer's needs. While an available agent is being chosen, the customer may be connected to a voice announcement system that informs the customer that he or she is
30 being connected to an agent and an approximate wait time. When ready, the CPC system 20 connects the customer to an agent (28, 30). While the customer is speaking to the agent, the CPC system 20 may monitor the call.

INSA

 \sqrt{A} \sqrt{A}

By directly acknowledging the Invite request with the Trying and In-Band Ringing responses sent at 204 and 206, the CPC system 10 acts as a SIP server. Thus, generation and communication of the Trying and In-Band responses are controlled by the UAS 106 in the CPC system 20.

Next, in its client mode, the CPC system 20 (under control of the UAC 104) then sends an Invite request (at 208) to the IVR system. The Invite request sent at 208 includes an origination address X associated with the user system 16, a destination address I associated with the IVR system 22 (Fig. 1), and a via address S corresponding to the address of the CPC system 20. Upon receipt of the Invite request, the IVR system 22 returns a Trying response (at 210) followed by a 183 In-Band Ringing response (at 212). The Trying and Ringing responses are absorbed by the UAC 104. Once it has accepted the call, the IVR system 22 sends an OK response (at 214) to the UAC 104 in the call processing module 100. In response, under control of the proxy 102 in the call processing module 100, an OK response is sent (at 216) back to the user system 16 to indicate that the Invite request (sent at 202) has been accepted.

The OK response sent by the IVR system 22 contains a message body (in SDP format) that describes the capabilities of the IVR system 22. The message body is included in the OK response sent (at 216) to the user system 16. In response, the user system 16 sends an Ack request (at 218) back to the UAS 106 in the CPC system. The proxy 102 in the CPC system 20 forwards the Ack request (at 220) to the IVR system 22. At this point, a call has been established (at 222) between the user system 16 and the IVR system. The media path for the call 222 is between the user system and IVR system directly (without passing through the CPC system 20).

During the IVR call 222, the IVR system 22 may prompt the user to input requested information. The IVR system 22 can collect voice data, DTMF (dual tone multi-frequency) signals, or other types of input data. At the conclusion of the IVR call, after the IVR system has collected the desired user information (e.g., user name, account information, and so forth), the IVR call 222 may be released. This is accomplished by the transmission of a Bye request (at 224) from the IVR system 22 to the CPC system 20. The CPC system 20 returns an OK response (at 226) to indicate that the Bye request has been successfully processed. The IVR call 222 at this point has terminated. However, the call session involving the calling user system and the community 26 is not released. Instead, the CPC system 20 will attempt to establish another call in the call session between the user system and another entity in the community 26.

The CPC system 20 may then send (at 228) an Invite request to an appropriate agent system based on the collected customer information. For example, the customer

may have bought a certain type of product which is handled by a predetermined department. The origination address in the Invite request at 228 is the address X of the user system 16, the destination address A corresponds to the agent system, and the via field is the address S corresponding to the CPC system 20 acting in its proxy capacity. Upon receipt of the Invite request, the agent system returns a Trying response (at 230) followed by a 183 In-Band Ringing response (at 232). While the CPC system 20 is re-connecting the call, it may play some pre-recorded message.

To re-connect the call session from the IVR system 22 to the agent system, the CPC system 20 sends an Invite request (at 234) to the user system 16 to redirect the media path to be between the user system 16 and the agent system (instead of the IVR system 22). The Invite request includes a message body that contains information describing the agent system. To indicate that this is part of the same call session with the call user system, the Invite request includes the same call ID with a higher call sequence number. The user system 16 returns an OK response (at 236) when it accepts the Invite request sent at 234. In response, the CPC system 20 sends an Ack request to the user system.

The agent system sends an OK response (at 240) back to the CPC system 20 to accept the Invite request sent at 228. A reliable response message scheme may be used to insure that the OK response sent at 240 is after the re-invite sequence performed at 234, 236 and 238. One example of a reliable response message scheme is described in IETF Internet Draft, entitled "Reliability of Provisional Responses in SIP," dated May 1999. Upon receipt of the OK response at 240, the CPC system 20 returns an Ack request (at 241) to establish a call (at 242) between the user system and the agent system.

To terminate the call, the agent system may send a Bye request (at 244) back to the CPC system 20. The CPC system 20, in its proxy capacity, then sends the Bye request (at 246) to the user system. The user system returns an OK response (at 248), with the CPC system 20 forwarding an OK response (at 250) to the agent system to acknowledge the Bye request sent at 246.

Referring to Fig. 4, a state diagram represents processing of various messages performed by the call processing module 100 (Fig. 2). The state diagram of Fig. 4 tracks events occurring in the message flow diagram of Fig. 3. The UAS 106 in the call processing module 100 waits (at 300) for receipt of an incoming call request from

a user system. When an Invite request is received, the UAS 106 generates (at 302) response messages to send back to the calling user system (204 and 206 in Fig. 3).

Next, the UAC 104 in the call processing module 100 generates (at 304) an Invite request to send to the IVR system 22 (208 in Fig. 3). The UAC 104 then processes (at 306) response messages received from the IVR system 22 (210 and 212 in Fig. 3). Upon receipt of an OK response (214 in Fig. 3) from the IVR system 22, the proxy 102 in the call processing module 100 generates an OK response (at 308) to send to the user system (216 in Fig. 3). When an Ack request is received (sent at 218 in Fig. 3), the proxy 102 generates an Ack request (at 310) to send to the IVR system 22 (220 in Fig. 3). At this point, a call has been established (at 312), with the media path set up between the user system and the IVR system 22.

When the call with the IVR system 22 is terminated by a Bye request from the IVR system 22, the UAS 106 generates (at 314) an OK response to send to the IVR system 22 (226 in Fig. 3). The call processing module 100 then proceeds to re-connect the user system with another element in the community 26, which may be one of the agent systems. To do so, the UAC 104 generates an Invite request (at 316) to send to the agent system (228 in Fig. 3). The UAC 104 processes (at 318) response messages from the agent system (230 and 232 in Fig. 3). Next, the UAC 104 generates an Invite request (at 320) to send to the user system (234 in Fig. 3). The UAC 104 also exchanges messages with the user system and the agent system (236, 238, and 240 in Fig. 3) to establish the call between the user system and the agent system (at 322).

Upon receipt of a Bye request from the agent system to terminate the call session, the proxy 102 generates (at 324) a Bye request to send to the user system. Upon receipt of an OK response from the user system, the proxy 102 generates an OK response to the agent system to acknowledge the Bye request.

A system that combines several logical entities, including a client, server, and proxy, is used for processing calls and directing the calls to appropriate ones of plural destinations. Such destinations may be part of a predefined community, such as a call center. The community may be associated with a single or a relatively small number of identifiers, such as telephone numbers. When a caller from outside the community communicates a call request containing one of the identifiers associated with the community, the call processing system responds initially as if it was the intended destination. The call processing system then proceeds to gather data about the caller

and to identify an appropriate one of the destinations within the community that should be connected to the user system. Once a connection is established, a media path provided directly between the user system and the appropriate one of the destinations within the community is established. By combining tasks performed by various logical entities for call processing, flexibility may be enhanced. The combined system allows internal management (within the community) of how inbound calls are processed and which destinations are accessed internally in response to a given inbound call. Thus, a single or a relatively small number of identifiers may be presented to the outside world while internally, multiple parties (greater than the number of identifiers presented to the outside world) are present.

The various software modules, routines, or other layers discussed herein may be executed on control units, such as one or more control units 118 shown in Fig. 2. Each control unit may include a microprocessor, a microcontroller, a processor card (including one or more microprocessors or microcontrollers), or other control or computing devices. The control unit may be coupled to one or more storage units, which may include one or more machine-readable storage media for storing data and instructions. As used here, a "controller" may refer to software, hardware, or a combination of both.

The storage media may include different forms of memory including semiconductor memory devices such as dynamic or static random access memories (DRAMs or SRAMs), erasable and programmable read-only memories (EPROMs), electrically erasable and programmable read-only memories (EEPROMs) and flash memories; magnetic disks such as fixed, floppy and removable disks; other magnetic media including tape; and optical media such as compact discs (CDs) or digital video discs (DVDs). Instructions that make up the various software routines, modules, or other layers may be stored in respective storage units. The instructions when executed by a respective control unit cause the corresponding node or element to perform programmed acts.

INV A² *Ⓢ* The instructions of the software routines or modules may be loaded or transported into the node or element in one of many different ways. For example, code segments including instructions stored on floppy discs, CD or DVD media, a hard disk, or transported through a network interface card, modem, or other interface device may be loaded into the system and executed as corresponding software routines or modules. In the loading or transport process, data signals that are

5 While the invention has been disclosed with respect to a limited number of
embodiments, those skilled in the art will appreciate numerous modifications and
variations therefrom. It is intended that the appended claims cover all such
modifications and variations as fall within the true spirit and scope of the invention.